RAD with JBoss AS

Rails : Sinatra : Rack : Grails : Roo : Play!

Agenda

- Use Case Overview
- Ruby – Lance
- Grails – Marius
- Roo – Marius
- Play! – Burr

Use Case

- Build a Twitter clone in 3 easy steps!
- Write it in something other than Java
- Deploy it

TorqueBox + Ruby

- Ruby on JBoss AS6
- Infinispan cache
- Quartz scheduling
- Tomcat web
- HornetQ messaging
- PicketBox authentication
- mod_cluster clustering

We've Got No Time, So What Do We Look At?

- Controller & routes: application.rb
- Domain models: lib/chirp.rb
- Views: views/layout.haml
- Deployment: config/torquebox.yml
- http://github.com/torquebox/chirpr

Pros & Cons

Pros

- You got your Java in my Ruby (CDI)

```
@bean = inject('SomeMCBean')
```

- Large & growing Ruby community
- YAML Deployment
- Hot application reloading
- Testing – test your Java code with Ruby
- HAML + SASS
- Dynamically typed

Pros & Cons

Cons

- Dynamically typed language
- Different programming paradigm, e.g.
  - Module mixins vs. class hierarchies
  - Code blocks
- Speed/Performance?

Play! Framework

www.playframework.org

Burr Sutter – Sr Product Manager, JBoss

Pros & Cons

- Pros:
  - Java Language (as a dynamic language via Eclipse compiler)
  - Edit, Save, Refresh – hot compile
  - Built-in Test harness – Unit, Functional, Selenium
  - RESTful Architecture
  - Custom Tags & Extensions
- Cons:
  - Less mature
  - Smaller community
  - Maven support is a recent addition (Ivy built-in)
  - Unique template system based on Groovy

Play! on JBoss

- “play war –o \tmp\my.war” Play 1.1.1
- jboss-web.xml
- Deploy a Datasource
- Tweak application.conf (datasource)
- “XAResource” error – remove jta1.1.jar
- Remove slf4j-api-1.6.1.jar
- Add hibernate-validator-legacy.jar
- Remove .settings, test-result, eclipse

Grails

(Née “Groovy on Rails”)

Grails

- Convention-over-configuration web framework
- Dynamic language-based (Groovy)
- Since 2006

Technology

- Entity-Controller-View division
  - Entity: Hibernate (GORM)
  - Controller: Spring MVC
  - View: JSP/GSP
- Coding in Groovy
- Plugins for extending functionality
  - Scripts for generating code

Pros & Cons

- Pros
  - Concise (CoC + dynamic language)
  - Mature (plugins)
  - JVM-based (investment reuse, performance)
- Cons
  - Learning curve for specific constructs
  - Not easy to reverse action done by scripts
  - Non-transparent dependency bundling

Spring Roo

(Kanga **Roo)**

Spring Roo

- Productivity tool for Spring applications
- Java-based
  - Focus on static typing
- No runtime components
  - Generates code, but has no libraries
- Command-line driven

Changes needed?

- Code works out-of-the box
- Adding complex use cases require manual coding
- Persistence infrastructure is local
  - Can be changed to managed/JTA manually
- Dependencies need some adjustment
- Suppress scanning

Pros / Cons

- Pros
  - Fast generation of Spring configuration
  - Fast generation of entity & scaffolding code
  - Good separation of generated/custom code
- Cons
  - Less flexible infrastructure generation
    - Exclusive focus on servlet containers
  - More limited set of options and templates
    - Compared with Grails, for example